The Exercise

The Gaussian Hill

Euler and Adams-Bashforth Schemes:

In Convection-Diffusion, the stability condition depends on the value of Pe:

$$if\ Pe \leq \sqrt{3}; C \leq \frac{Pe}{3}$$
$$if\ Pe > \sqrt{3}; C \leq \frac{1}{Pe}$$

The implementation of forward Euler method is done. For the first time step a very small time step is chosen. Following this Adams Bashforth scheme has also been employed. By varying the Peclet number (Pe) and the Courant number (C), various cases has been observed.

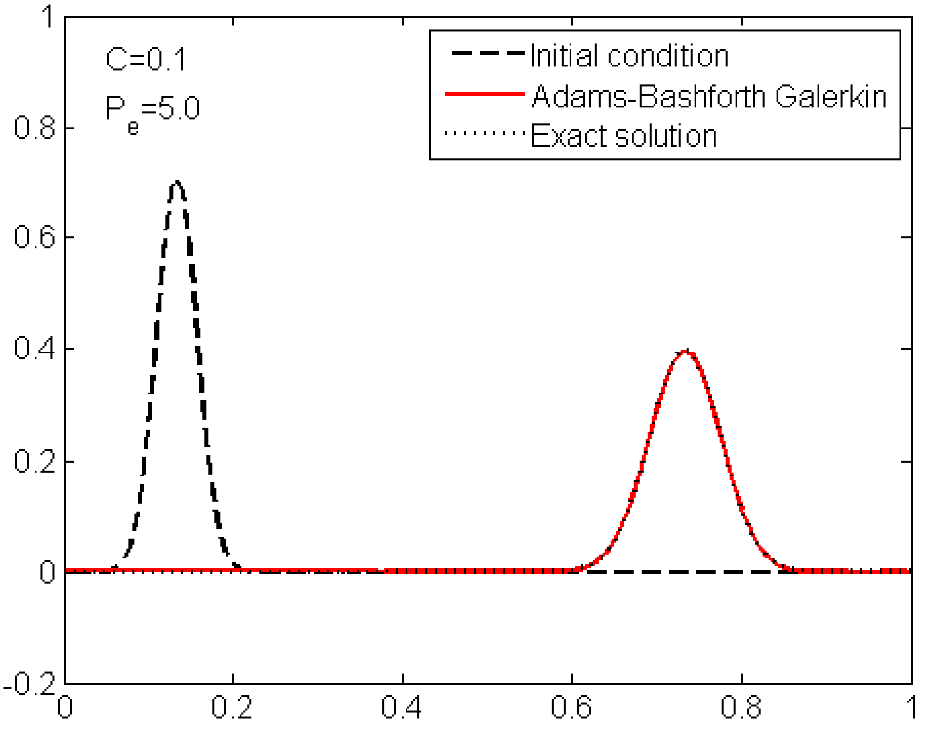

Fig 1: First case, comparison of the solution obtained using the Adams-Bashforth method for C=0.1 and Pe=5 ,for which the method behaves well

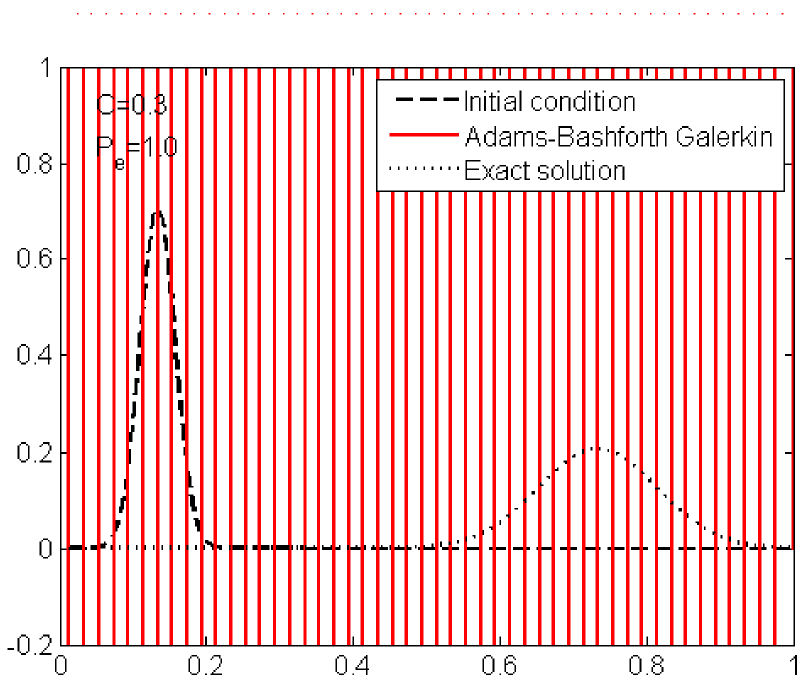

Fig 2: Second Case , comparison of the solution obtained using the Adams-Bashforth method for C=0.3 and Pe=1, for which the case becomes unstable as C is comparatively higher for the Pe=1

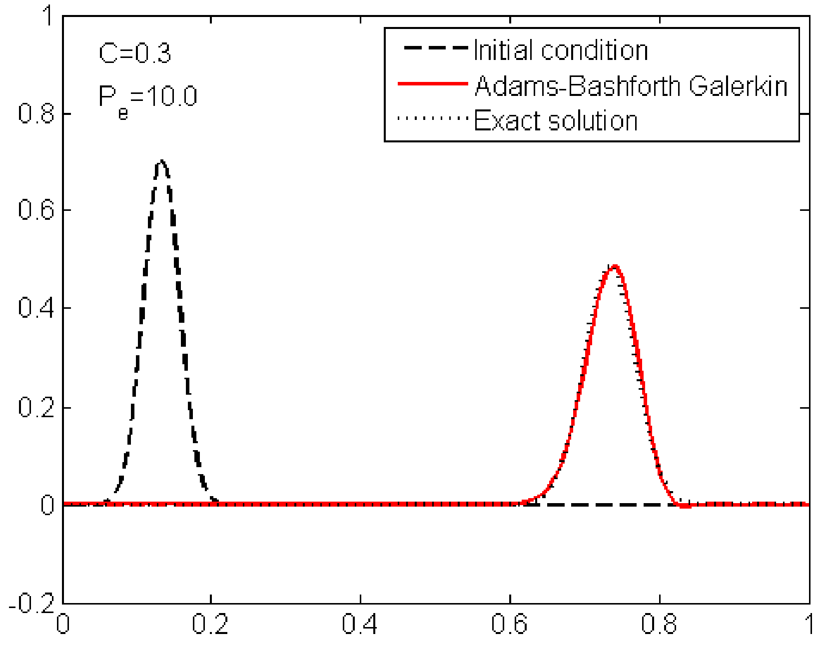

Fig 3: Third Case, comparison of the solution obtained using the Adams-Bashforth method for C=0.3 and Pe=10 , where method behaves well

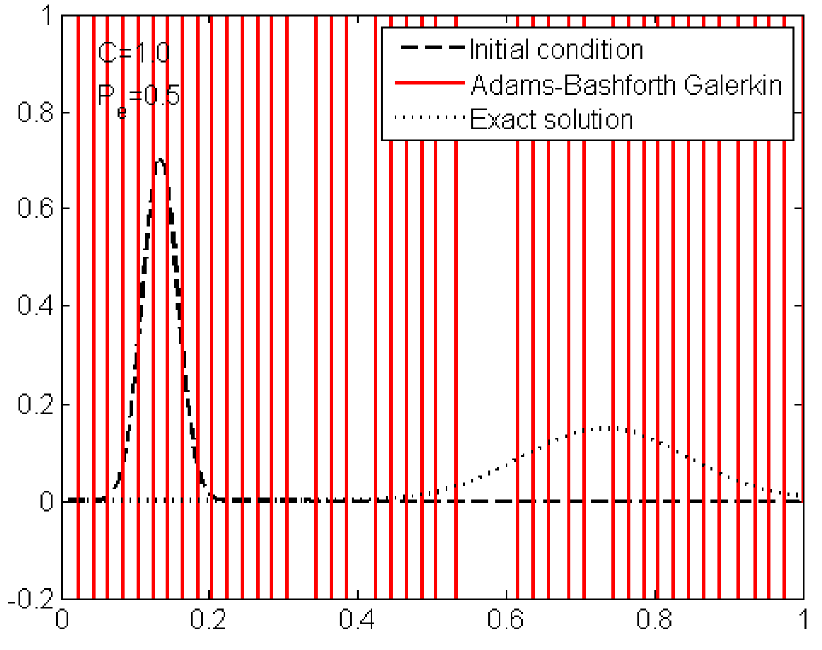

Fig 4: Forth case, Comparison of the solution obtained using the Adams-Bashforth method for C=1 ad Pe=0.5, where the unstable solution is obtained.

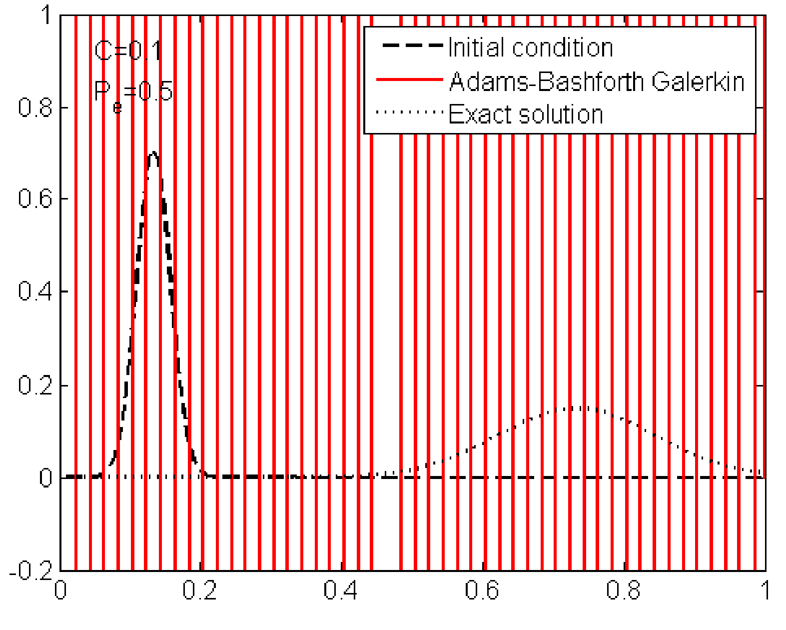

Fig 5: Fifth case, Comparison of the solution obtained using the Adams-Bashforth method for C=0.1 and Pe=0.5, unstable solution as Pe is low instead of low C.

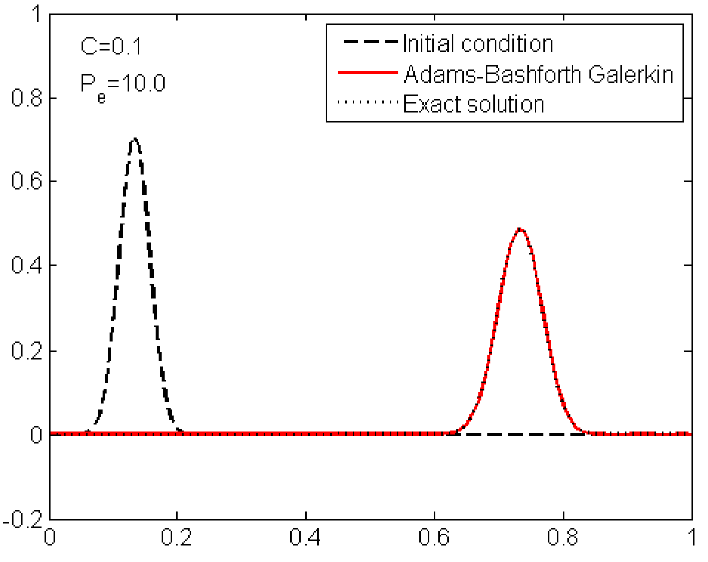

Fig 6: Sixth case, comparison of the solution obtained using the Adam-Bashforth method of the C=0.1 and Pe=10, accurate solution as Pe is comparatively higher and C is low.

When C is increased, Adams-Basforth method does not behave well. A few more cases have been considered for C=3, Pe=1; C=3, Pe=5; C=3, Pe=100; C=4, Pe= 100. For all these cases Adams- Basforth method gives unstable solutions as C is very high; instead of values of Pe.

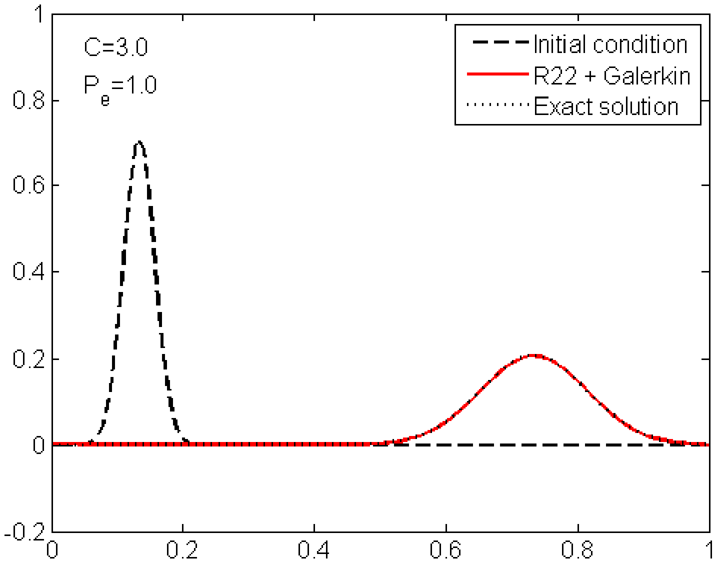

Fig 7: Comparison of the solution obtained using the $R_{2,2}$ for C=3 and Pe=1 with the exact solution

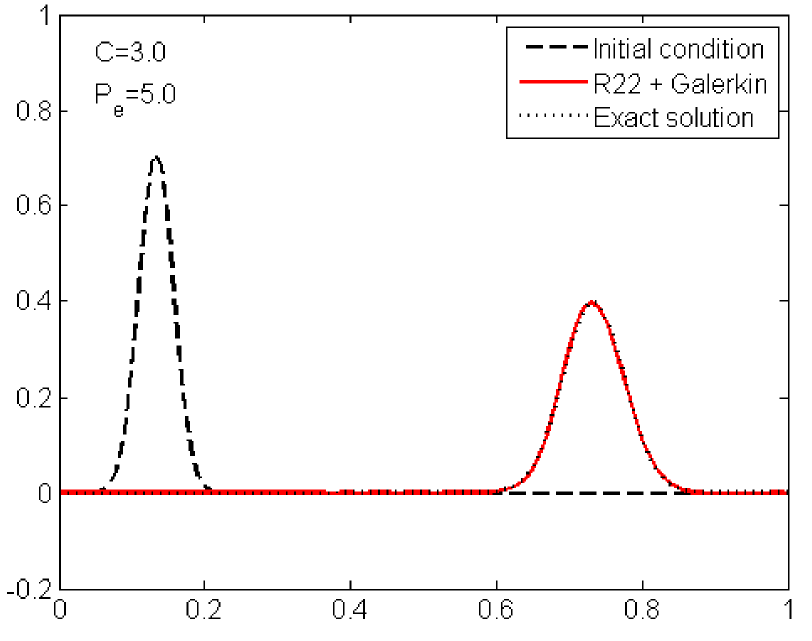

Fig 8: Comparison of the solution obtained using the $R_{2,2}$ for C=3 and Pe=5 with the exact solution

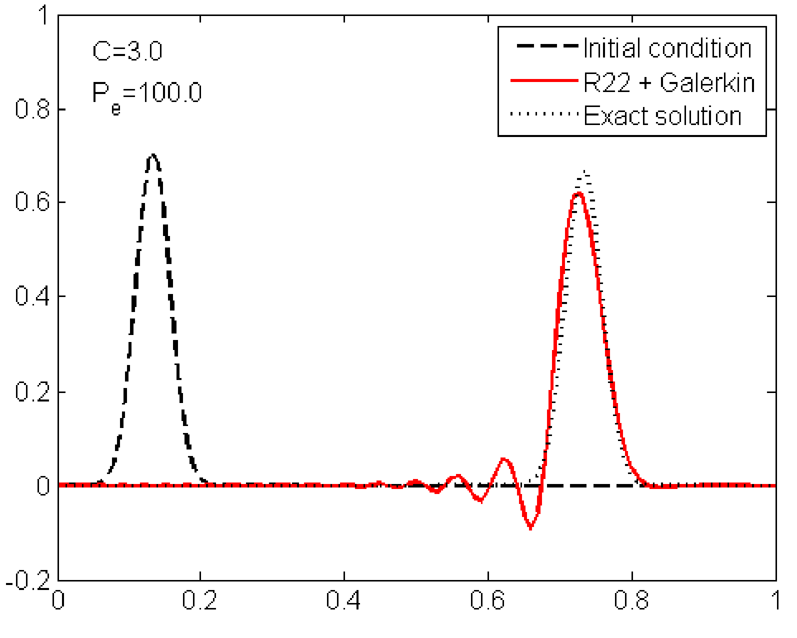

Fig 9: Comparison of the solution obtained using the $R_{2,2}$ for C=3 and Pe=100 with the exact solution

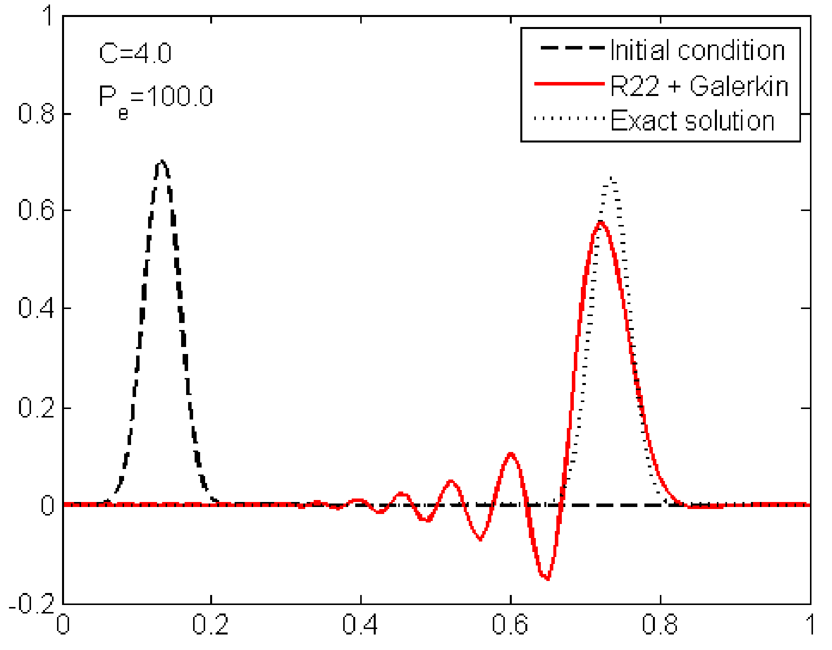

Fig 10: Comparison of the solution obtained using the $R_{2,2}$ for C=4 and Pe=100 with the exact solution

Pade $R_{2,2}$ Scheme:

For 1: C=3, Pe=1; 2: C=3, Pe=5; 3: C=3, Pe=100; 4: C=4, Pe=100 the behavior of the method has been studied. The method behaves well in-comparison to the Adams-Bashforth Method , at high C values,. It can be observed from the Figs 7,8,9, and 10; where for these cases the Adams-Bashforth was unstable for the same Pe, and C values. Figs 9 and 10 shows relative errors in amplitude and phase for the fourth-order method. $R_{2,2}$. The $R_{2,2}$ scheme gives improved accurate results for later two cases as shown in the figs 12 and 14.

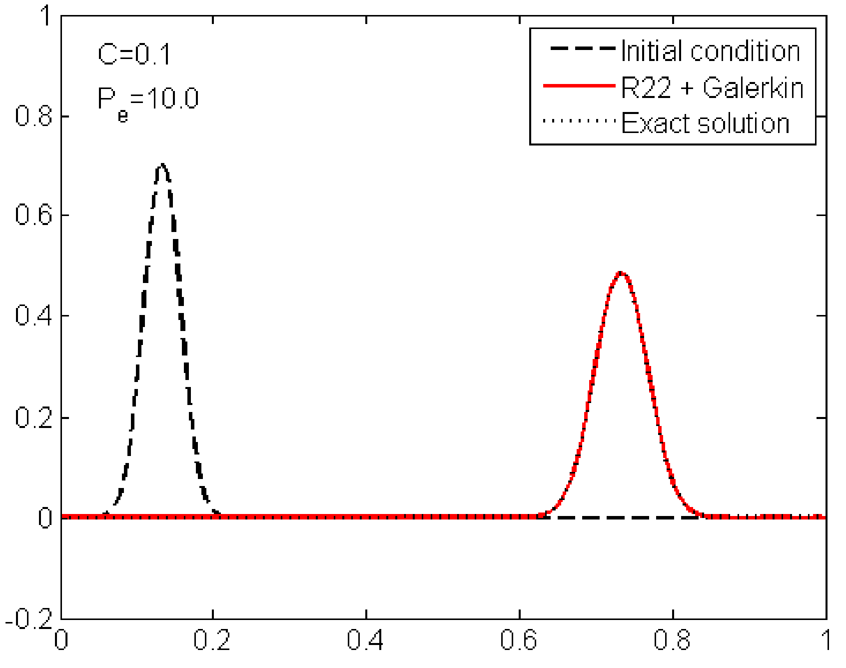

Fig 11: Comparison of the solution obtained using the $R_{2,2}$for C=0.1 and Pe=10 with the exact solution

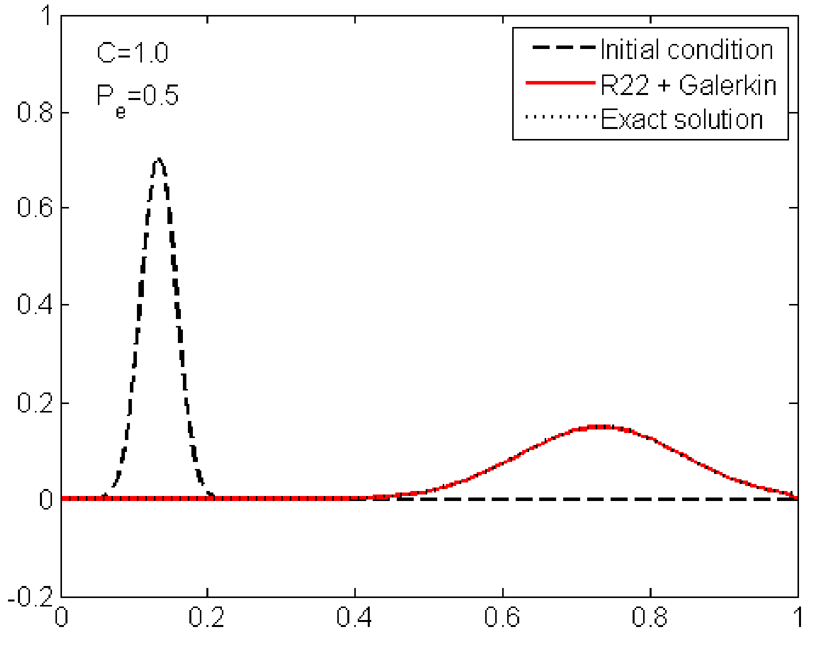

Fig 12: Comparison of the solution obtained using the $R_{2,2}$ for C=1 and Pe=0.5 with the exact solution

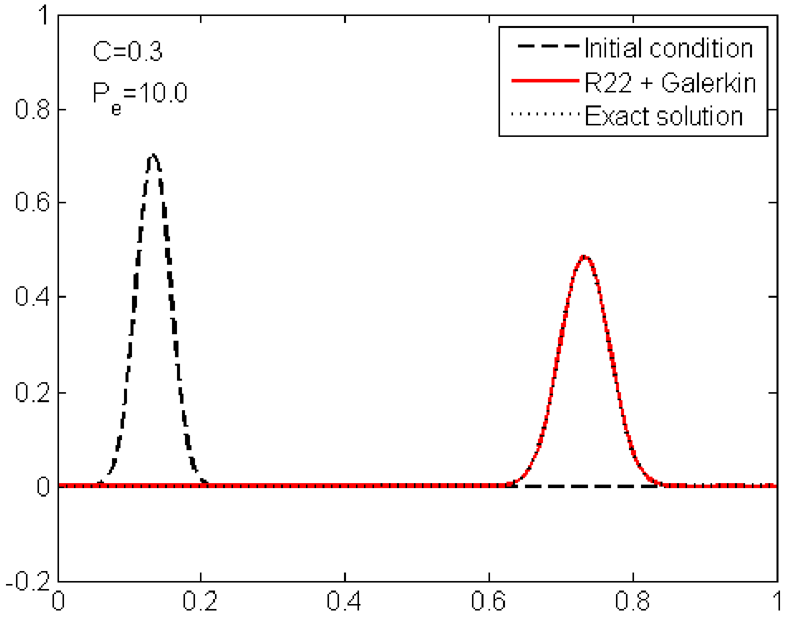

Fig 13: Comparison of the solution obtained using the $R_{2,2}$ for C=0.3 and Pe=10 with the exact solution

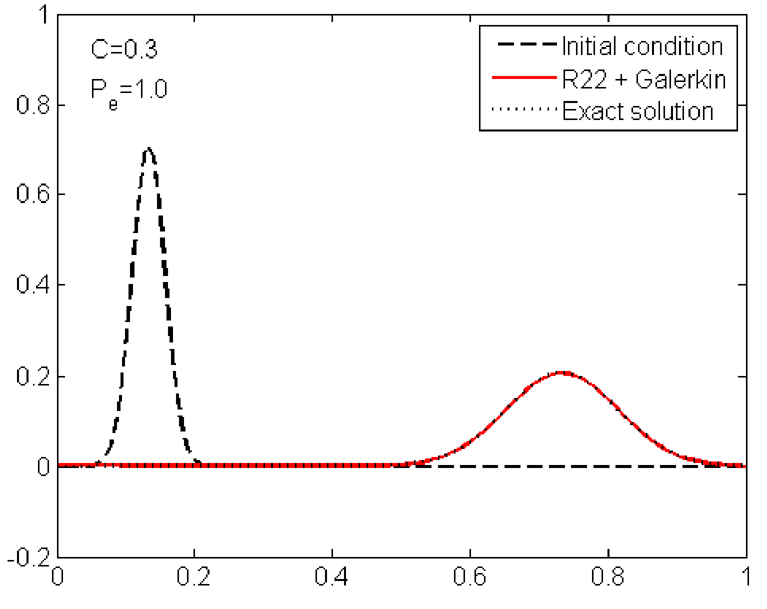

Fig 14: Comparison of the solution obtained using the $R_{2,2}$ for C=0.3 and Pe=1 with the exact solution

Adams Bash-forth Scheme:-

```
disp('There are four integration schemes available ')
disp(' [0] = Crank-Nicolson');
disp(' [1] = R22');
disp(' [2] = R33');
disp(' [4] = Adam-Basforth');
%disp(' [4] = Forward-Euler');
disp('and five methods to perform spatial discretization')
disp(' [0] = Galerkin');
disp(' [1] = Least-Squares');
disp(' [2] = Streamline-Upwind Petrov-Galerkin (SUPG)');
disp(' [3] = Galerkin Least-Squares (GLS)');
disp(' [4] = Sub-Grid Scale (SGS)');
disp(' ')
d_temp = input('Choose a method to perform time integration = ');
d_esp = input('and another one for the spatial discretization = ');
```

```
elseif d_temp == 4
method = 'Adams-Bashforth ';
W = 0;
w = 1;%%%% Start with Euler; Will change to Adams-Basforth in
Galerkin1
else
error('Unavailable time integration scheme')
end

% Solution
if d_esp==0 && d_temp==4
Sol = Galerkin1(W,w,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 0 && d_temp~=4
Sol = Galerkin(W,w,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 1
Sol = ILS(W,w,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 2
Sol = SUPG(W,w,tau,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 3
Sol = GLS(W,w,tau,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
elseif d_esp == 4
Sol = SGS(W,w,tau,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd,bccd);
end
```

`function Sol =Galerkin1(T,s,a,nu,f,K,M,G,xnode,dt,nstep,c,Accd1,bccd1)`

Time step loop:

```
% Loop to compute the transient solution
for p=1:nstep
if p==1 %%%Forward Euler method loop
aux = dt*(-Kt*c + Mf);
F = [];
for i =1:n
F = [F; s(i)*aux];
end
F = [F;bccd*0];
dc = U\(L\F);
dc = reshape(dc(1:n*npoin),npoin,n);
c = c + sum(dc,2);
Sol = [Sol c];
s=1.5*s; %%%%w=1.5 for Adam basforth
else
```

```
aux = dt*(-Kt*c+ Mf);
F = [];
for i =1:n
F = [F; s(i)*(aux)+0.5*Kt*dt*Sol(:,p-1)];%%%% for Adam basforth
end
F = [F;bccd*0];
dc = U\(L\F);
dc = reshape(dc(1:n*npoin),npoin,n);
c = c + sum(dc,2);
Sol = [Sol c];
end
aux = dt*(-Kt*c+ Mf);
F = [];
for i =1:n
F = [F; s(i)*(aux)+0.5*Kt*dt*Sol(:,p-1)];%%%% for Adam basforth
end
F = [F;bccd*0];
dc = U\(L\F);
dc = reshape(dc(1:n*npoin),npoin,n);
c = c + sum(dc,2);
Sol = [Sol c];
end
```